

## **Proyecto: Gas & Food**

### **Plan del proyecto del software**

**(Formato Pressman)**

#### **Miembros del equipo:**

An Wei Pham Luo	Alejandro Pascua Piña
Andrea Martínez Fernández	Beatriz Villegas Sánchez
Darío Fernando Gallegos Quishpe	Enrique Salazar del Cid
Javier Martín-Pozuelo Salvador	Jennifer Marmolejos Urbáez
Juan Martín Bárez Alonso	Manuel Oreja Valverde
Pedro Pablo Doménech Arellano	

## Control de cambios

Número de versión	Fecha	Autores	Descripción
1	08/12/2016	Andrea , Bea, Darío	Rellenar campos
2	09/12/2016	Pedro P.	5.1
3	11/12/2016	Pedro P.	5 (5.2 incompleto)
4	12/12/2016	Pedro P.	5 (últimos retoques)
5	12/12/2016	An W., Enrique S.	6.
6.	12/12/2016	Alex, Andrea	4.
7.	13/12/2016	Beatriz V., Andrea	3. , 4.
8.	13/12/2016	Manuel, Javi	2
9.	15/02/2017	Alex, Manu	2 y 4
10.	18/02/2017	Pedro P., Manuel, Alex, Juan, Andrea	
11.	21/02/2017	Manu	1 (modelo de proceso)

# Índice

1.Introducción	6
1.1Propósito del plan	6
1.2Ámbito del proyecto y objetivos	6
1.2.1 Declaración del ámbito	6
1.2.2 Funciones principales	6
1.2.3 Aspectos de rendimiento	6
1.2.4 Restricciones y técnicas de gestión	6
1.3Modelo de proceso	6
2.Estimaciones del proyecto	7
2.1Datos históricos	7
2.2Técnicas de estimación	7
2.3Estimaciones de esfuerzo, coste y duración	7
3.Estrategia de gestión del riesgo	8
3.1Análisis del riesgo	8
3.2Estudio de los riesgos	8
3.3Plan de gestión del riesgo	8
4.Planificación temporal	9
4.1Estructura de descomposición del trabajo o Planificación temporal	9
4.2Gráfico Gantt	9
4.3Red de tareas	9
4.4Tabla de uso de recursos	9
5.Recursos del proyecto	10
5.1Personal	10
5.2Hardware y software	10
5.3Lista de recursos	10
6.Organización del personal	11
6.1Estructura de equipo (si procede)	11
6.2Informes de gestión	11
7.Mecanismos de seguimiento y control	12
7.1Garantía de calidad y control	12
7.2Gestión y control de cambios	12
8.Apéndices	
13	



## **1. Introducción**

En este documento detallamos el plan de desarrollo de nuestro proyecto. Especificando la distribución de las tareas entre los distintos miembros del grupo y los procesos de implementación teniendo en cuenta los requisitos establecidos por el cliente.

### **1.1 Propósito del plan**

El propósito del plan de proyecto es establecer y precisar los objetivos y modelos de organización de la aplicación. Para efectuar todas las fases del desarrollo conociendo el dominio de la aplicación y sus metas.

### **1.2 Ámbito del proyecto y objetivos**

#### **1.2.1 Declaración del ámbito**

**Nuestra aplicación se gestiona de la siguiente manera:**

- Gestión de gasolineras. Contendrá el listado de las gasolineras disponibles y sus precios actuales.
- Gestión de tiendas de alimentación. Funciona igual que la gestión de gasolineras.
- Gestión de usuarios. Almacena, actualiza y borra los usuarios registrados en el fichero de usuarios.
- Gestión de sugerencias. Las sugerencias de los usuarios se almacenan en un fichero al que los administradores tienen acceso y permiso para eliminarlas.
- Gestión del uso del mapa
- Otras gestiones. Uso del mapa, comparador, carga de archivos de ficheros, mapa y relevancia.

#### **1.2.2 Funciones principales**

#### **1.2.3 Aspectos de rendimiento**

#### **1.2.4 Restricciones y técnicas de gestión**

### **1.3 Modelo de proceso(TODOS)**

Se utiliza el modelo unificado. Hemos elegido este modelo de proceso porque funciona bien con proyectos modulares como éste además de trabajar con casos de usos, los cuáles ya hemos realizado. Realizaremos varias iteraciones en cada módulo, las cuales constan de diseño y codificación-pruebas. Además distinguiremos las 4 fases del modelo:

-Inicio: haremos las primeras iteraciones del módulo más sencillo como toma de contacto con el proyecto.

-Elaboración y construcción: partiendo de casos de uso ya diseñados (elaboración) realizaremos una iteración en el resto de módulos para construirlos (construcción).

-Adaptación: nos aseguraremos con pruebas exhaustivas de que todos los módulos contruidos y terminados funcionen correctamente juntos y por separado

## **2. Estimaciones del proyecto**

### **2.1 Datos históricos**

Como datos históricos se toma como referencia un proyecto de la asignatura Ingeniería del software de años previos con características similares a nuestro proyecto.

En concreto, este proyecto consistía en la gestión de un buscador de productos informáticos. Duró 7 meses y fue desarrollado por un equipo de 6 personas.

El código se implementó en JAVA y contó con unas 1000 líneas de código.

### **2.2 Técnicas de estimación**

Ámbito del software:

Esta aplicación será utilizada por el usuario para obtener información comparativa de los distintos precios que ofrecen distintas tiendas de alimentación y gasolineras más cercanas a una ubicación. Además cuenta con un mapa en el que se podrá localizar dichos locales, el cual ayudará al usuario a identificar la ubicación de estos.

Por otro lado, la aplicación tendrá un sistema de sugerencias que permitirá al usuario poder notificar a los administradores de la aplicación sobre posibles cambios en los precios de tiendas de alimentación.

Finalmente se cuenta también con un sistema de promociones aplicables en las tiendas que muestra la aplicación, la cuales sólo serán usadas por aquellos usuarios registrados, mientras que aquellos no registrados se limitarán a usar el buscador/comparador/mapa.

Descomposición basado en el problema:

\*Datos de estimación -> Esperado = (optimista + 5 x realista + 2 x pesimista)/8

Tarea	Optimista	Pesimista	Realista	Esperado	Coste
Gestión de gasolineras	100	200	150	156	1872
Gestión de tiendas de alimentación	150	275	200	212	2544
Gestión de usuarios	150	275	200	212	2544
Gestión de gasolinas	80	120	100	102	1224
Gestión de alimentos	80	120	100	102	1224
Uso del buscador y comparador	250	325	280	285	3420
TOTAL				1069 LDC	12828€

Descomposición basado en el proceso:

	Diseño	Codificación	Pruebas	Total
Módulo gasolineras	7	4	7	18
Módulo tiendas alimenta	7	4	7	18

ción				
<b>Módulo Usuarios</b>	7	4	7	18
<b>Módulo gasolinas</b>	6	3	6	15
<b>Módulo alimentos</b>	6	3	6	15
<b>Módulo búsquedas y comparación</b>	9	6	9	24
<b>TOTAL</b>				108 personas x hora

\*El esfuerzo se estima en personas\*hora.

\*Utilizaremos LDC como variable de estimación (basado en la cantidad).

\*Tarifa: 12€/LDC.

\*El coste se estima en Euros (€).

### 2.3 Estimaciones de esfuerzo, coste y duración

Para la estimación del esfuerzo no utilizaremos ninguna técnica parametrizada debido a la falta de experiencia para calcular ciertas constantes de estos métodos de estimación de esfuerzo.

\*Recursos de equipo: 11 personas.

-Estimación basada en el problema

Productividad = 70 LDC/(Mes x persona) es una unidad basada en la cantidad de líneas capaces de hacer una persona en un mes trabajando 8h mensuales (2 horas semanales). Con la siguiente fórmula estaremos el esfuerzo en



personas x mes basándonos en la anterior unidad y las LDC totales del proyecto.

$$\text{Esfuerzo} = \sum (\text{LDC totales del proyecto}) / (\text{LDC}/(\text{Mes x persona})) = 1069/70 = 15,2 \text{ personas x mes}$$

Por otro lado el coste ya está calculado en la tabla superior (14940€). De esta forma cobraremos por el precio de nuestras LDC ya que no habrá costes en nuevas herramientas de desarrollo u otros.

Duración(teórica): 15,2 personas x mes / 11 personas que forman el equipo de desarrollo = 1,4 meses (diseño, código y pruebas)

-Estimación basada en el proceso

$$\text{Esfuerzo} = 108 \text{ personas x hora} = 13,5 \text{ personas x mes}$$

$$\text{Duración(teórica)} = 13,5 \text{ personas x mes} / 11 \text{ personas} = 1,22 \text{ meses}$$

### **3. Estrategia de gestión del riesgo**

#### **3.1 Análisis del riesgo**

El proyecto será supervisado desde un primer momento para poder evitar futuros riesgos. Eso nos ayudará a que los riesgos que en un principio no se tuvieron en cuenta, y que podrían llegar a ser problemáticos para el funcionamiento y desarrollo de este, estén bajo control y en la lista de planificación.

Por consiguiente, en el proyecto se acomodará a la estrategia proactiva, la cual nos servirá durante el desarrollo de este, para solucionar eficientemente y con rapidez los problemas que vayan surgiendo y que se planificaron con antelación antes de ponerlo en marcha.

Además, detallaremos una serie de planes de contingencia, que se irán ampliando con el análisis continuo de los riesgos a medida que se desarrolla el proyecto.

#### **3.2 Estudio de los riesgos**

En el estudio de riesgo nos encargamos de planificar las posibles contingencias. Para nuestra aplicación existen varios tipos: riesgos relacionados con el proyecto, en concreto con la planificación de este, del personal y de los requisitos; los riesgos técnicos que pueden frenar el desarrollo del software, como lo son los requisitos y la incertidumbre a la hora de implementar.

A medida que avance el proyecto necesitaremos concretar ciertas tareas las cuales acarrearán otras más complejas necesarias para el desarrollo del proyecto. Por lo que la línea de desarrollo general o planificación acabará por dividirse y moderarse para efectuar estos cambios.

A lo largo del proyecto podría darse el caso de que alguno de los miembros lo abandonara, obligando al resto de integrantes a asumir la parte del trabajo que les correspondía. Esto

provocaría un retraso general por lo que no se cumpliría la planificación temporal establecida.

Por último, en la relación de riesgos del proyecto, los requisitos están basados en la idea de que es un proyecto que no acabará, en un principio, con salida al mercado, por lo que muchos de los requisitos que se encuentran en el SRS no serán suficientes y , por otro lado, estos tampoco son tan concretos como deberían ser a la hora de crear la aplicación, ya que, se ajusta a la capacidad de trabajo y conocimientos de los integrantes del proyecto, que todavía es muy limitado.

Otro tipo de riesgos a tener en cuenta son los técnicos, primordialmente por nuestra capacidad de programación y de visión del proyecto. A la hora de implementar el desarrollo software de nuestra aplicación habría que considerar la programación gráfica, que a estas alturas del proyecto no podemos definir. De nuevo, los requisitos del software no están del todo concretados para poder desarrollar de forma correcta la aplicación.

### **3.3 Plan de gestión del riesgo**

#### **3.3.1.Introducción**

Para llevar a cabo esta gestión se procederá a la identificar, analizar y priorizar los riesgos que pueden surgir durante el desarrollo del proyecto.

Una vez identificados los riesgos más prioritarios, se les aplicará el plan RSGR, el cual es el conjunto de planes de reducción, supervisión y gestión del riesgo.

#### **3.3.2. Identificación de riesgos.**

Para identificar los riesgos del proyecto, hemos utilizado la "*Boehm's Top 10 Software Risk Items*".

##### **1.Deficiencia del personal.**

- a)Organización errónea.
- b)Falta de información/ comunicación.

##### **2.Planificación y presupuesto poco realistas.**

- a)Planificación errónea.
- b)Presupuesto muy bajo.

##### **3.Desarrollo erróneo de la implementación de la aplicación.**

- a)Prototipado mal diseñado.
- b)Usuario insatisfecho con las funciones implementadas.

##### **4.Mal desarrollo de la interfaz de usuario.**

- a)Interfaz poco amigable.

### 5.Cambio continuo de requisitos.

a)Cambio de requisitos base/ Aumento de trabajo.

### 6.Deficiencias de componentes externos

a)Diseño de la aplicación en un lenguaje en desuso.

b)Diseño de la aplicación que no es compatible con el rendimiento de los componentes.

c)Diseño de una aplicación que no es capaz de adaptarse a los cambios continuos del software.

### **3.3.3. Análisis y priorización de riesgos del proyecto**

La siguiente tabla de riesgos se ha ordenado primero por la probabilidad de que el riesgo se haga real y segundo por la gravedad de las consecuencias. Hemos asignado las probabilidades y consecuencias de cada riesgo basándonos en la tabla SQAS-SEI:

TABLA DE RIESGOS
<p><b>1º.Prototipado mal diseñado:</b> la idea de la aplicación, los requisitos y su función final tiene que estar clara desde el principio hasta el final sino daría problemas a la hora de crear ciertos matices de la aplicación, las entradas de estas.</p> <p><b><u>Probabilidad:</u></b> Probable, por nuestro defectuoso conocimiento sobre la programación. <b><u>Consecuencia:</u></b> Crítica, el stakeholder podría rechazar la aplicación final o intermedia obligandonos a reedificar.</p>
<p><b>2º.Planificación errónea:</b> provocando un retraso de la entrega o realización del proyecto.</p> <p><b><u>Probabilidad:</u></b> Probable, debido a nuestra falta de experiencia en ello. <b><u>Consecuencia:</u></b> Crítica, llevaría tiempo reorganizar las entregas para poder mantener el hilo de los demás trabajos.</p>
<p><b>3º.Organización errónea:</b> disminuiría la calidad del proyecto, además de aumentar el tiempo que habría que invertir en amortiguarlo.</p> <p><b><u>Probabilidad:</u></b> Probable, debido al poco conocimiento de las directrices básicas de organización. <b><u>Consecuencia:</u></b> Crítica, ya que presuponemos que la organización ya realizada es correcta y sería una corrección de la actual.</p>
<p><b>4º.Presupuesto muy bajo:</b> si fuese un proyecto para una empresa que no aportase suficiente dinero para realizarlo, acabaría provocando un proyecto de baja calidad e ineficiente.</p>

**Probabilidad:** Remota, no es probable ya que lo económico es lo que manda.

**Consecuencia:** Catastrophic, si no hay fondos económicos para realizar la aplicación no se podría entregar el proyecto y se iniciarán acciones legales.

**5º. Diseño de la aplicación que no es compatible con el rendimiento de los componentes:** Si no desarrollamos la aplicación para que funcione en los dispositivos actuales que tienen un cierto rendimiento, de forma general podría suponer el fin para este producto y por tanto ser inservible para aquellos usuarios con dispositivos de alto/bajo rendimiento.

**Probabilidad:** Ocasional, por lo general se trabajan con equipos asequibles, es decir, con unos componentes hardware bastante genéricos y actuales.

**Consecuencia:** Crítica, si no se comprueba en dispositivos tecnológicos ínfimos para conocer su rendimiento.

**6º. Falta de información/ comunicación:** disminuiría la calidad del proyecto, provocaría la repetición del material incluido en el proyecto.

**Probabilidad:** Frecuente, cada intervención sobre el proyecto nos llevaría confrontar ideas.

**Consecuencia:** Severa, es frecuente que debido a la realización de pocas reuniones u a la mala organización, el equipo no esté bien informado y por lo tanto la aplicación no se podrá desarrollar de forma correcta.

**7º. Diseño de una aplicación que no es capaz de adaptarse a los cambios continuos del software:** las aplicaciones sólidas robustas tienen que abarcar los posibles cambios que hay en los dispositivos donde se usará lo que acarrea una extra planificación, la cual debe cubrir los posibles cambios posibles.

**Probabilidad:** Probable, para ello habría que tener versiones de la aplicación.

**Consecuencia:** Severa, la idea de la creación software es que sea viable en un largo periodo de tiempo por lo que si en meses se queda obsoleta el stakeholder podría llevarse una mala imagen de la cooperativa.

**8º. Interfaz poco amigable:** Habría que hacer un estudio para establecer el público al que va dirigida. Y comprobar si es fácil de manejar por este público, porque para cierto rango de edades podría ser un poco complejo y no tendría una buena acogida en el mercado.

**Probabilidad:** Ocasional, por lo general la línea de gustos entre los stakeholder suele ser lineal, no tendría por qué generar algún disgusto.

<p><b><u>Consecuencia:</u></b> Severa, no es la parte más compleja del desarrollo software por lo que no llevaría mucho tiempo en arreglar, si el tema es estético</p>
<p><b>9º. Usuario insatisfecho con las funciones implementadas:</b> que el usuario no tenga claras las funciones de su aplicación acarrea retraso y pérdidas económicas innecesarias. Además, si no se nos proporciona la información necesaria para realizar una buena implementación de la aplicación, más tarde se convertirá en un problema de aceptación de la aplicación por parte de este.</p> <p><b><u>Probabilidad:</u></b> Ocasional, el trato con el stakeholder es fluido por lo que cada adelanto sería en confrontada comunicación con este.</p> <p><b><u>Consecuencia:</u></b> Severa, se debería mejorar o rehacer la aplicación para asegurar la satisfacción del usuario con la implementación.</p>
<p><b>10º. Cambio de requisitos base/ Aumento de trabajo:</b> Una mala definición de requisitos desde el comienzo del proyecto, puede provocar serios problemas, ya que serían la base del proyecto y si se cambian habría que empezar de cero. También puede darse el caso de que durante el proyecto se realicen pequeños cambios en los requisitos ya existentes y esto conllevaría un retraso en la planificación temporal, ya que aumentaría el trabajo de los integrantes del equipo.</p> <p><b><u>Probabilidad:</u></b> Frecuente, el bajo conocimiento sobre el desarrollo de una aplicación y sobre la definición de requisitos ocasiona el olvido de ciertos matices y restricciones necesarias.</p> <p><b><u>Consecuencia:</u></b> Menor, si es un cambio durante el desarrollo. Catastrófica, si es un cambio en los requisitos base.</p>
<p><b>11º. Diseño de la aplicación en un lenguaje en desuso:</b> Se estudiará el mercado para ver cuáles son las opciones más viables para que el producto no quede obsoleto y se venda.</p> <p><b><u>Probabilidad:</u></b> Improbable, los lenguajes de programación suelen seguir una línea estable de reconocimiento por lo que sería complicado que lenguajes muy usados en la actualidad y que están en la mayoría de los dispositivos actuales dejen de funcionar o usarse de un mes para otro.</p> <p><b><u>Consecuencia:</u></b> Severa, debido a que habría que adaptar la aplicación al nuevo lenguaje y si no tienen canales de comunicación visibles podría tardar meses o años en rehacerse.</p>

### 3.3.4. Reducción, supervisión y gestión del riesgo (Plan RSGR)

A continuación procedemos a dar un plan RSGR para los riesgos más prioritarios, obtenidos de la tabla de riesgos:

<b>Organización errónea</b>	
<b>Reducción</b>	Para evitar ciertos riesgos y una situación de descontrol como por ejemplo, en el reparto de tareas a cada integrante, la jerarquía a seguir, la falta de integrantes y comunicación, se establecerá la organización de equipo más favorable y se irá observando como avanza el proyecto.
<b>Supervisión</b>	Se comprobará a mediados del mes que se ha establecido esa organización, que la planificación se cumple y que las tareas se han desarrollado correctamente.
<b>Gestión del riesgo</b>	Si la organización establecida no funciona y no permite que se avance en el desarrollo del proyecto, se realizará un cambio de organización y una reestructuración de la planificación.

<b>Planificación errónea</b>	
<b>Reducción</b>	Se podría dar el caso de que debido a ciertos retrasos en algunas tareas críticas, se retrasara todo el proyecto y esto nos obligaría a retrasar la entrega. Pero hay que entregarlo dentro del plazo (realista), por lo que si no tenemos en cuenta estos posibles retrasos todo el proyecto peligraría.
<b>Supervisión</b>	Comprobar semanalmente que el proyecto cumple con la planificación temporal.
<b>Gestión del riesgo</b>	Se podría dar el caso de que se retrasara una tarea crítica, la cual retrasa todo el proyecto, y que al comprobar la planificación comprobamos que no se podría entregar el proyecto en fecha. Por lo que realizaremos una nueva planificación para redistribuir el trabajo y así entregar el proyecto dentro del plazo.

<b>Prototipado mal diseñado</b>	
<b>Reducción</b>	Cada mes se intercambiarán los códigos del programa de la aplicación entre los distintos integrantes del grupo. Así será más fácil detectar posibles errores de funcionamiento e incluso podrán encontrar soluciones a algunos problemas, surgidos por la falta de conocimiento del lenguaje de programación u a la falta de información, que se hayan encontrado los otros integrantes.

<b>Supervisión</b>	Se realizará de la manera anteriormente citada, durante todo el proyecto. Observaremos cómo avanza la implementación del proyecto.
<b>Gestión del riesgo</b>	Si se les hubiera escapado algún error grave en la implementación, que es algo muy común, se realizaría una revisión en conjunto, con todos los miembros.

## 4. Planificación temporal

- Cada semana o cada dos semanas se comprobará mediante la realización de un pequeño test, si todos los integrantes del plan de proyecto tienen las mismas ideas en cuanto a este.
- Se dará un plazo de un mes al jefe de proyecto para comprobar cómo funciona el equipo con él y si el reparto de tareas es equitativo y eficaz.
- Cada semana se comprobará la ausencia de los integrantes.
- Cada mes los integrantes se intercambiarán entre ellos los códigos de implementación.

### 4.1 Estructura de descomposición del trabajo o Planificación temporal

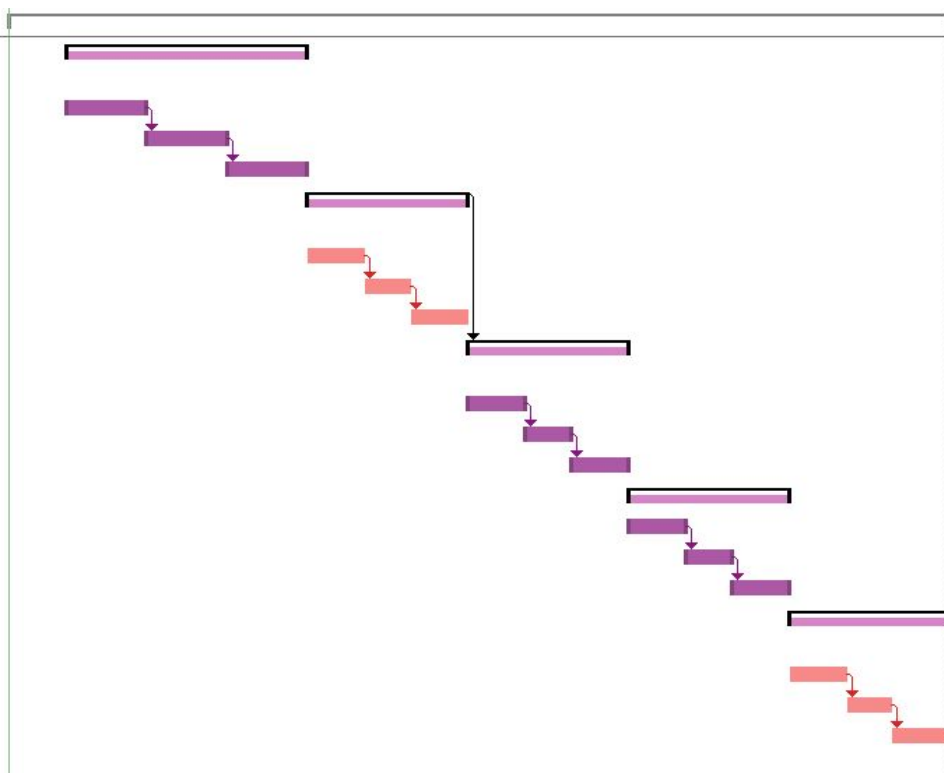
	<b>Requisitos</b>	<b>SRS</b>	<b>Plan de proyecto</b>	<b>Diseño</b>	<b>Codificación</b>	<b>Pruebas</b>
<b>Módulo Descuento</b>	Todos 10/10/16 - 30/11/16	Todos  2/11/16 - 20/4/17	Todos 08/12/2016 - 20/4/2017	Todos Iteración 1: 27/2/17 - 1/3/17  Iteración 2: 6/3/17 - 8/3/17  Iteración 3: 13/3/17- 15/3/17	Todos Iteración 1: 2/3/17 - 3/3/17  Iteración 2: 9/3/17 - 10/3/17  Iteración 3: 16/3/17- 17/3/17	Todos Iteración 1: 4/3/17 - 5/3/17  Iteración 2: 11/3/17 - 12/3/17  Iteración 3: 18/3/17- 19/3/17

<b>Módulo Productos</b>	Todos 10/10/16 - 30/11/16	Todos 2/11/16 - 20/4/17	Todos 08/12/2016 - 20/4/2017	Todos 20/3/17 - 24/3/17	Todos 25/3/17 - 28/3/17	Todos 29/3/17 - 2/4/17
<b>Módulo Establecimiento</b>	Todos 10/10/16 - 30/11/16	Todos 2/11/16 - 20/4/17	Todos 08/12/2016 - 20/4/2017	Todos 3/4/17 - 7/4/17	Todos 8/4/17 - 11/4/17	Todos 12/4/17 - 16/4/17
<b>Módulo Usuarios</b>	Todos 10/10/16 - 30/11/16	Todos 2/11/16 - 20/4/17	Todos 08/12/2016 - 20/4/2017	Todos 17/4/17 - 21/4/17	Todos 22/4/17 - 25/4/17	Todos 26/4/17 - 30/4/17
<b>Módulo comparación y búsqueda</b>	Todos 10/10/16 - 30/11/16	Todos 2/11/16 - 20/4/17	Todos 08/12/2016 - 20/4/2017	Todos 1/5/17 - 5/5/17	Todos 6/5/17 - 9/5/17	Todos 10/5/17 - 14/5/17

## 4.2 Gráfico Gantt



Proyecto	81 días?	mié 22/02/1	dom 14/05/1
1 Módulo Descuento	21 días	lun 27/02/17	dom 19/03/17
1.1 Iteración 1	7 días	lun 27/02/17	dom 05/03/17
1.2 Iteración 2	7 días	lun 06/03/17	dom 12/03/17
1.3 Iteración 3	7 días	lun 13/03/17	dom 19/03/17
2 Módulo Productos	14 días	lun 20/03/17	dom 02/04/17
2.1 Diseño	5 días	lun 20/03/17	vie 24/03/17
2.2 Codificación	4 días	sáb 25/03/17	mar 28/03/17
2.3 Pruebas	5 días	mié 29/03/17	dom 02/04/17
3 Módulo Establecimiento	14 días	lun 03/04/17	dom 16/04/17
3.1 Diseño	5 días	lun 03/04/17	vie 07/04/17
3.2 Codificación	4 días	sáb 08/04/17	mar 11/04/17
3.3 Pruebas	5 días	mié 12/04/17	dom 16/04/17
4 Módulo Usuario	14 días	lun 17/04/17	dom 30/04/17
4.1 Diseño	5 días	lun 17/04/17	vie 21/04/17
4.2 Codificación	4 días	sáb 22/04/17	mar 25/04/17
4.3 Pruebas	5 días	mié 26/04/17	dom 30/04/17
5 Módulo Búsqueda	14 días	lun 01/05/17	dom 14/05/17
5.1 Diseño	5 días	lun 01/05/17	vie 05/05/17
5.2 Codificación	4 días	sáb 06/05/17	mar 09/05/17
5.3 Pruebas	5 días	mié 10/05/17	dom 14/05/17



### 4.3 Red de tareas

#### Anexo 1

### 4.4 Tabla de uso de recursos

	Septiembre	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo
Personal	11	11	11	11	11	11	11	11	11
Ordenadores	11	11	11	11	11	11	11	11	11
Gmail	✓	✓	✓	✓	✓	✓	✓	✓	✓
Google Drive	✓	✓	✓	✓	✓	✓	✓	✓	✓
Google Docs	✓	✓	✓	✓	✓	✓	✓	✓	✓
Eclipse	✗	✗	✗	✗	✗	✓	✓	✓	✓
Microsoft Project	✗	✗	✗	✓	✓	✓	✓	✓	✓

## **5. Recursos del proyecto**

### **5.1 Personal**

Se dispone para el desarrollo de este proyecto de un equipo formado por once personas. Cada uno de los miembros tiene el deber de asegurar que el proyecto se lleva a cabo cumpliendo con los plazos de entrega del mismo, tanto realizando el propio trabajo como advirtiendo al resto de miembros de la no realización de alguna tarea en caso de ser necesario. Por tanto, las decisiones importantes serán tomadas por todos los miembros de forma consensuada.

### **5.2 Hardware y software**

Se dispone de al menos una computadora por cada integrante del equipo ya sea portátil o sobremesa. Estas computadoras trabajan con diversos sistemas operativos: Windows 7, Windows 8.1, Windows 10 y Ubuntu 14.04. Cuentan además con unas características hardware de como mínimo gama media.

Además de las computadoras personales el equipo tiene acceso a los diferentes laboratorios de la Facultad de Informática de la UCM, los cuales están provistos de computadores e impresoras.

A esto se le han de sumar otros dispositivos como son tablets y smartphones.

Como recursos software se tiene a disposición de todo el equipo las diversas herramientas de Google como son Google Drive, Gmail y Google docs.

Para la planificación del proyecto se usará la herramienta de Microsoft project.

Para la elaboración de la documentación se tienen plantUML, docker, IBM RSA.

Para llevar a cabo de las tareas de programación contamos con el entorno de desarrollo de Eclipse

### **5.3 Lista de recursos**

Computadoras (portátiles y sobremesa), smartphones, tablets, impresoras, servicios de Google, material de oficina (papel, bolígrafo, etc), paquete de Microsoft Office, plantUML, docker, IBM RSA.

## **6. Organización del personal**

## 6.1 Estructura de equipo (si procede)

El equipo Gas & Food está formado por 11 personas, y se utilizará la estructura Mantei DC (Descentralizado Controlado), en el cual se asignará un jefe que será el encargado de organizar y tomar las decisiones más relevantes para el equipo, y varios subjefes para las diferentes subtareas.

La estructura del equipo estará compuesta por grupos de 2 o 3 personas, cuyo tamaño podría variar dependiendo de la cantidad de carga de trabajo y horas correspondientes al proyecto. Asignados por cada funcionalidad, nos organizaremos de esta forma para poder centrar nuestra dedicación eficientemente.

Existe conexión y dependencia entre los grupos, de tal manera que si surge algún problema relacionado con el proyecto puedan ser capaces de ayudarse mutuamente.

Comenzaremos en cuatro módulos:

- **Módulo de establecimientos:**
    - Manuel Oreja Valverde(jefe y subjefe).
  - **Módulo de productos:**
    - Beatriz Villegas Sánchez(subjefe).
    - Jennifer Marmolejos Urbáez
  - **Módulo de usuarios:**
    - Alejandro Pascua Piña(subjefe).
    - An Wei Pham Luo.
    - Pedro Pablo Doménech Arellano.
    - Andrea Martínez Fernández.
  - **Módulo de búsqueda:**
    - Javier Martín-Pozuelo Salvador(subjefe).
    - Darío Fernando Gallegos Quishpe.
    - Enrique Salazar del Cid.
- Juan Martín Bárez Alonso(pendiente de asignar)

## **6.2 Informes de gestión**

Utilizamos el SRS como documentación importante a la hora de realizar las distintas funcionalidades que tendrá el proyecto. De esta forma, haremos reuniones y memorias sobre los últimos cambios y novedades que vamos realizando en el proyecto, además de contemplar cambios entre los grupos de trabajo y solventar dudas generales. Cada grupo se enfocará en su trabajo asignado, asimismo en tareas que requieran de todos los componentes del equipo.

Cada grupo realizará una memoria sobre los cambios realizados recientemente en su trabajo asignado y además de los problemas encontrados, se notificará si se han solventado y de qué manera.

En las reuniones de trabajo se juntarán las memorias para notificar al equipo de las novedades y para su propia discusión. También cada componente del equipo dará su opinión del transcurso del desarrollo del trabajo y se llegará a un acuerdo.

En estos encuentros se realizará una propuesta de objetivos futuros a tomar en los grupos y en el equipo para el correcto progreso del proyecto.

## **7. Mecanismos de seguimiento y control**

### **7.1 Garantía de calidad y control**

El objetivo primordial será proporcionar visibilidad sobre el proceso del desarrollo. La calidad del programa se garantizará mediante técnicas de Ingeniería del Software. Durante el desarrollo se repartirá el trabajo entre los miembros intentando asignar el más apto a cada uno.

Además se cuenta con un grupo en una aplicación de mensajería que agilizará la resolución de dudas simples siendo ya recomendable una pequeña reunión en el caso de una mayor complejidad.

Se realizarán revisiones técnicas formales (RTF) y se auditará la aplicación para asegurar que esta sigue dentro de los estándares requeridos. Las RTFs nos ayudarán a seguir un desarrollo correcto y a verificar que la aplicación alcanza los requisitos del SRS.

Estas RTFs se realizarán a cada hito del proyecto o entre hitos si el desarrollo de estos se alarga más de lo planeado. Tanto los miembros que hayan ayudado a alcanzar ese hito como los miembros que vayan a desarrollar módulos del programa dependientes de estos deberían estar presentes. Estos últimos deben prepararse la reunión para poder comprobar si los requisitos del paso anterior se encuentran satisfechos. Las reuniones no deben alargarse a más de una hora en la medida de lo posible.

### **7.2 Gestión y control de cambios**

Para el desarrollo de este documento se ha usado Google Drive. Este servicio ha permitido la edición de documentos de manera simultánea, un control de cambios y un espacio para compartir ficheros. La experiencia ha sido correcta y se planea seguir usando este servicio para compartir documentos e información complementaria al desarrollo de la aplicación durante el mismo.

El desarrollo de la aplicación se gestionará mediante Git. Esta herramienta cuenta con un potente control de cambios y es distribuida, permitiendo que varios miembros puedan trabajar a la vez y/o continuar el trabajo de otros sin muchas trabas.

Durante el desarrollo se incentivarán diferentes buenas prácticas para facilitar y acelerar el desarrollo:

- Acordar un flujo de trabajo con los miembros que vayan a trabajar en el mismo código
- Comentar tanto el código en sí como cada versión de un fichero que se suba
- Los cambios deben centrarse en el mismo módulo o función del proyecto
- No se deben subir módulos que produzcan errores
- Probar siempre el código antes de subir un cambio

En el caso en el que una tarea de garantía de calidad y control falle y un defecto del programa llegue a etapas avanzadas de su desarrollo, se seguirá el siguiente proceso:

- Se analizará el impacto del defecto
- Se priorizará su corrección de manera acorde al análisis
- Se propondrá una corrección

Dependiendo de la fase del desarrollo en la que se encuentre la aplicación en el momento de proponer este cambio, este afectará en menor o mayor medida al resto de componentes ya desarrolladas. En este punto se evaluará la aptitud de ese cambio y su aprobación dependerá de un consenso entre los miembros del grupo.

## **8. Apéndices**

Información sobre calidad del software:

ACIMED 3(3):40-42, septiembre-diciembre, 1995

[http://www.bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)